Conference Reports

In this issue:

Every two years, 75–100 systems, database, and application developers and researchers gather at Asilomar for the Workshop on High Performance Transaction Processing Systems (HPTS). The name, something of a misnomer, stems from its origin in 1985, when Jim Gray, Dieter Gawlick, Andreas Reuter, and other luminaries invited practitioners and academics to discuss the challenges and successes in the area of large, scalable, high-throughput systems. Today, I think of HPTS as the place where people with large-scale problems come to talk with people who like to solve large-scale problems. The crowd is a seamless blend of researchers and practitioners and infrastructure suppliers and consumers.

Each HPTS seems to have a dinstinctive flavor. A few years back, it seemed that all the large online service providers were talking about how they used Lucene to solve large-scale problems. This year, there was a lot of talk about integrating NoSQL solutions into large-scale services. HPTS feels a lot like HotOS, but with more emphasis on data and less emphasis on operating systems. This year Rik Farrow went to HPTS to soak in the ambience, learn a bit about the community, and coordinate student scribes so that we could bring a taste of HPTS to the USENIX community. Unlike many USENIX workshops, HPTS is not based on paper submissions; the written record mostly consists of personal blog postings, a collection of presentations, and some less-than-one-page submissions. These reports are the closest thing you'll find to an HPTS proceedings, although Web surfing will reveal several personal blog reports.

—Margo Seltzer, USENIX Acting Executive Director

14th International Workshop on High Performance Transaction Systems (HPTS)

Pacific Grove, CA
October 23–26, 2011

Datacenter Trends 101

Summarized by Eugenia Gabrielova (eugenia.g@uci.edu)

Internet-Scale Datacenter Economics: Where the Costs & Opportunities Lie

James Hamilton, Amazon

James kicked off HPTS by saying it is his favorite conference, primarily because of the people in the room. He then claimed there has been more innovation in the past five years than in the previous fifteen, primarily due to advances in cloud computing and the accessibility it provides to application developers. Datacenters are expensive and don't really help innovation—when you are spending millions or billions of dollars, you do things the way you know it will work.

At Amazon, there are always multiple datacenters under construction. In the past four years, AWS has evolved into a phenomenal business generating tons of revenue and passing on savings to customers. Amazon was approximately a $2.7 billion annual revenue enterprise in 2000. Now, every day Amazon Web Services adds enough capacity to have support ed all of Amazon.com's infrastructure in the company's first five years. There is a competitive advantage in having better infrastructure.

The talk shifted to everything below the OS, because that is generally where the money goes. Charts often show people costs, but at a really large scale these costs are very minor relative to the costs of servers and power distribution. As a rule of thumb, "If you want to show people your infrastructure, you're probably spending too much." In the monthly costs of a datacenter, servers (not power distribution) dominate. However, server costs are decreasing, while networking costs are creeping up. Networking is a problem precisely because it is "trending up," so it is broken—this is a huge opportunity for innovation.

Another area with great potential for innovation is cooling systems, which have remained the same for about 30 years. Fans moving air is expensive, and moving water is also fairly expensive. Datacenters of the future could be designed beautifully with eco-cooling, no AC required. In the meantime, modular and pre-fabricated datacenters are regaining popularity, because of how quickly they can be deployed. Making datacenters better isn't just a technical advantage, it is an enormous business advantage.

Bruce Lindsay (Independent, ex-IBM) commented on the declining cost of network ports. Someone asked about OpenFlow, and James said that Google supports Quagga for routing, and OpenFlow comes from Stanford. Both open up the infrastructure by allowing the control plane to run centrally, with cheap hardware for running the data plane.

Someone noted that standard practice in the computer industry is to "prepare for the worst." James replied that there are test sites running with high-voltage direct current and many high-profile datacenters have very robust strategies for ensuring uptime (such as fully dedicated power generators). However, due to high demand, it can be hard to know which workloads will be running in a datacenter at a given time.

Slides from this talk can be found at http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_HPTS2011.pdf. James can be reached at James@amazon.com.

The Rise of Dark Silicon

Nikos Hardavellas, Northwestern University

Dr. Hardavellas was unable to make it to HPTS this year but has made the slides for his talk available at www.hpts.ws/sessions/Hardavellas.pdf.

The Hitchhiker's Guide to Precision Time Synchronization

Krishna Sankar, Egnyte

Before he become Lead Architect at Egnyte, Krishna was a Distinguished Engineer at Cisco Systems. In his free time he enjoys working as a technical judge for FIRST LEGO League Robotics. He began his talk by emphasizing that time synchronization is different from time distribution. There is incredible value in offering time precision in an application. Ocean observatory networks, industrial automation, cloud computing, and many other fields would benefit. Time synchronization is also slowly finding its way into routers and blade server fabrics.

Krishna gave an overview of IEEE 1588 v2 PTP (Precision Time Protocol), which concerns the sub-microsecond synchronization of real-time clocks in components of a network distributed measure and control system. This capability is intended for relatively localized systems, like those often found in finance, automation, and measurement. The purpose of IEEE 1588 is simple installation, support for heterogeneous clock systems, and minimal resource requirements on networks and host components.

PTP uses a master/slave model to synchronize clocks through packets over unicast and/or multicast transport. It follows a simple protocol: master and slave devices enabled with PTP send messages through logical ports to synchronize their time. Of the five basic PTP devices, four are clocks. Each clock determines the best master clock in its domain, including itself. It is very difficult to achieve high precision, so some hardware-assisted time stamping can be used to help accuracy (which is more complex than it sounds). A few key lessons in working with PTP are that shallow, separate networks are preferable; anything too hierarchical will prove difficult to manage and synchronize. Accuracy depends largely on hardware and software abilities and interaction. Additionally, GPS satellite visibility is needed for the GMC (Grand Master Clocks, the most accurate).

Krishna closed by encouraging audience members to submit to ISPCS 2012, which will take place in San Francisco. Learn more at http://www.ispcs.org. A central theme of the Q&A was whether these time precision techniques are accessible to the average application developer. How can an average application, subject to layers of virtualization and delays, take advantage of precision timing? The main takeaway was that, with some planning, developers can certainly take advantage of advances in time synchronization. The slides for this talk can be found at http://www.hpts.ws/sessions/Synchronization.pdf.

Not Your Traditional Data Management Session

Summarized by Andy Pavlo (pavlo@cs.brown.edu)

Enterprise Supercomputing

Ike Nassi, SAP

Ike began with a harsh denunciation and lamentation about current enterprise computing hardware, which supports only a single TB of DRAM on a single motherboard. That limitation makes it difficult for servers to be used for enterprise computing systems, because they often have a much greater working set size. Ike strongly believes it is time to re-examine our current predilection for shared-nothing architectures and that the database research community should take advantage of developments in high-performance computing research from the past 25 years, which has favored a shared-everything architecture. Large memory systems on the scale required by SAP are simply not being built; thus Ike sought to create one himself.

Ike presented a new DBMS server architecture, currently under development at SAP, which uses a virtual shared-

everything paradigm built on a single rack cluster. In SAP's new system, the database executes on a single instance of Linux, while underneath the hood the ScaleMP hypervisor routes operations and data access requests over networking links (i.e., no shared buses) to multiple, shared-nothing machines. By masking the location of resources through a coherent shared-memory model, Ike argues that they are able to minimize the amount of custom work individual application developers have to do in order their database platforms.

The early morning audience was languid, but several skeptics, such as Margo Seltzer, were concerned that the data links between machines would not match the speed of DRAM. Ike assured these doubters that high-performance communication links such as InfiniBand would be sufficient for this system. He also remarked that the system currently does not support distributed transactions; thus there is no message passing needed between nodes. Roger Bamford (Oracle) asked, why divide the system into so many cores, and Ike replied that they need the RAM. Adrian Cockcroft asked how common failures are. Ike said that this is a lab test so far, and in thirty days there were no failures. Margo Seltzer said she loved this project, which reminded her of late '80s shared memory multiprocessor systems such as Encore. Ike said that unlike the early systems, which used busses, their system is using fast serial connections, and he suggested that people not be blinded by what happened in the past. Both Margo and James Hamilton wondered about the problem of having a NUMA architecture, especially when the ratio of "near" memory to "far" memory reaches 10 to 1. Ike said that he lied, that all memory is used as if it were L4 cache. Roger pointed out the cost of going to the cache coordinator, and Ike replied that identifying the location of memory has a constant cost.

Forget Locality

Randal Burns, John Hopkins University

Randal Burns, a systems research professor at Johns Hopkins, raised the issue that the canonical optimizations used in DBMS systems were insufficient to achieve high-performance data processing (i.e., > 1 million IOPS) on large and complex graph data sets. This is because any algorithm that must perform a scan of the entire data set or a random walk in the graph cannot take advantage of locality in the data. Thus, optimizations such as partitioning, caching, and stream processing are rendered impotent.

Randal then discussed ongoing work at Hopkins that seeks to understand the main bottlenecks that prevent modern systems from scaling to larger I/O operation thresholds. His work shows that low-level optimizations to remove lock contention and interference can improve throughput by 40% over file access through the operating system.

Margo Seltzer asked whether making certain assumptions about the physical layout of the graphs could be exploited. That is, could performance be improved if the system stored the data in a way that optimized for a particular processing algorithm? Randal responded that such techniques would be unlikely to work for attribute-rich graphs, since there is no optimal ordering. Roger suggested that he put the answer in their database and be done with it, eliciting laughter. Mike Ubell asked whether the cache was throttling IOPS, and Randal said yes, that there is lots of bookkeeping and page structures to manage. James Hamilton asked why not have the database use memory directly, and Randal said that is where they are going. They want to get away from local and global data structures. James pointed out that databases had already done this. Mohan asked about latches, and Randal replied that they want only locks that matter, such as a read lock on dentry and on mapping.

Someone suggested proper indexing, declaration of graph processing, having the database make decisions in advance. Randal replied that that is ground that has been trod before. Someone else pointed out that it seemed they were looking for storage memory that had DRAM-like characteristics. Randal agreed, saying that without a memory hierarchy his talk would be a no-op.

Flexible Hardware for Flexible Data Intensive Software

Arun Jagatheesan, Samsung

Arun Jagatheesan from Samsung shared his perspective on new hardware trends and configurations for big-data systems and supercomputing platforms. He was specifically focused on the flexibility of both the hardware systems (i.e., allowing administrators to configure the hardware) and the software platforms that they support (i.e., allowing users to execute variegated workloads). Arun began with an overview of the flash-based Gordon system that he helped to develop while at the San Diego Supercomputer Center in 2009. Arun said that the three main lessons that he learned from this project were (1) not all the configuration options that one needs are available in hardware, (2) there is a nebulous tradeoff between flexibility and performance, and (3) manufacturers, applications, users, and administrators are unprepared for new hardware.

From this, Arun then introduced his more recent work on Mem-ASI at Samsung. Mem-ASI is a memory-based storage platform for multi-tenant systems that is designed to learn the access patterns and priorities of applications and react to them accordingly in order to improve throughput. Such priorities could be either service-level hints from applications, service-level requests from the computing platform's infrastructure, or simply how the individual application accesses data. This additional information could be used by the

system for more intelligent scheduling and resource management. Arun believes that such a model could both improve performance and possibly reduce energy consumption.

James Hamilton said he could understand the power savings, but not the factor of 4 for performance gains. Arun said that the idea is that you can change something on the memory controller to change what is happening at the transport layer. James asked if this had to do with the number of lanes coming off the core, and Arun replied that it is not about lanes but about what you can do behind those lanes.

Mapping the NoSQL Space

Summarized by Aaron Elmore (aelmore@cs.ucsb.edu)

The NoSQL Ecosystem

Adam Marcus, MIT

Adam Marcus provided a brief history of the origins of NoSQL, beginning in the late 1990s with Web applications developed using open source database systems. Applications that saw increased load began wrapping stand-alone DBMSes to allow for sharding to achieve scale. Additionally, relational operations were removed and joins were moved to the application layer to reduce costly database operations. These modifications led to the creation of databases that went beyond traditional SQL stores and came to be referred to as Not Only SQL (NoSQL). With a plethora of recent NoSQL options, Adam lightheartedly introduced Marcus's Law, which tells us that the number of persistence options doubles every 1.5 years.

The majority of NoSQL stores rely on eventual consistency and are built using a key-based data model, sloppy schemas, single-key transactions, and application-based joins. However, exceptions to these properties were highlighted, including alternatives to data models, query languages, transactional models, and consistency. For example, while many NoSQL databases utilize eventual consistency, many alternatives exist, such as PNUTS's timeline consistency or Dynamo's configurable consistency based on quorum size. With a basic understanding of NoSQL properties, real-world usage scenarios were outlined.

Recently, Netflix has undergone a transition from Oracle to Cassandra, to store customer profiles, movie watching logs, and detailed customer usage statistics. Key advantages that motivated the migration include asynchronous datacenter replication, online schema changes, and hooks for live backups. More information about this migration is detailed in Adrian Cockcroft's paper at http://www.slideshare.net/adrianco/migrating-netflix-from-oracle-to-global-cassandra. Contrasting Cassandra, Facebook chose HBase for the new FB Messages storage tier, primarily due to difficulties in programming against eventual consistency. HBase also provides a simple consistency model, flexible data models, and simplified distributed data node management. MongoDB usage for Craigslist archival and Foursquare check-ins were briefly highlighted.

After detailing NoSQL databases and use cases, Adam presented takeaways for the database community. First, and most contentiously, is developer accessibility. Adam said that the ability of a programmer to set up and start using a NoSQL db really mattered. Bruce Lindsay (ex-IBM) strongly objected to the question on "whether first impressions made within five minutes of database setup and use matter." Margo Seltzer (Harvard) countered that a new generation of developers, who use frameworks such as Ruby on Rails, do make decisions on accessibility and that these developers should matter. Adam furthered the argument by claiming accessibility will matter beyond minutes in schema evolution, scaling pains, and topology modifications. Database development should also examine the ecosystem of reuse found in some NoSQL projects. This is exemplified in Zookeeper, LevelDB, and Riak core becoming reusable components for systems beyond their initial development. Lastly, the NoSQL movement espouses the idea of *polyglot persistence*, where a specific tool is selected for a task. Selecting various data solutions can create painful data consistency issues, as an enterprise's data becomes spread among disjoint systems.

In closing, Adam presented several open questions. These focused on data consistency, datacenter operational tradeoffs, assistance for scaling up, the ability to compare NoSQL data stores, and next-generation databases. A question by C. Mohan (IBM) about the need for standardization of a query language drew mixed reactions.

The Present and Future of Apache Cassandra

Jonathan Ellis, DataStax

Jonathan Ellis, of DataStax and a major contributor to the Apache Cassandra project, outlined developments in the recent version 1.0 release and goals for future Cassandra releases. Inspired by Google's BigTable and Amazon's Dynamo, Cassandra began as a project at Facebook before becoming an Apache incubator project. Cassandra's popularity is partly due to the ability for multi-master (and thus multi-datacenter) operation, linear scalability, tunable consistency, and performance for large data sets. Cassandra's user base today includes large companies such as Netflix, Rackspace, Twitter, and Gamefly.

For release 1.0 of Cassandra, leveled compaction was introduced to improve the reconciliation of multiversion data files. Advantages over the previous size-tiered compaction include improved performance due to lower space overhead and fewer average files required for read operations. Addition-